2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Robotic Classification of Divers' Swimming States using Visual Pose Keypoints as IMUs

Demetrious T. Kutzke[1], Ying-Kun Wu[2], Elizabeth Terveen[3], and Junaed Sattar[4]

***Abstract*— Traditional human activity recognition uses either direct image analysis or data from wearable inertial measurement units (IMUs), but can be ineffective in challenging underwater environments. We introduce a novel hybrid approach that bridges this gap to monitor scuba diver safety. Our method leverages computer vision to generate high-fidelity motion data, effectively creating a "pseudo-IMU" from a stream of 3D human joint keypoints. This technique circumvents the critical problem of wireless signal attenuation in water, which plagues conventional diver-worn sensors communicating with an autonomous underwater vehicle (AUV). We apply this system to the vital task of identifying anomalous scuba diver behavior that signals the onset of a medical emergency such as cardiac arrest—a leading cause of scuba diving fatalities. By integrating our classifier onboard an AUV and conducting experiments with simulated distress scenarios, we demonstrate the utility and effectiveness of our method for advancing robotic monitoring and diver safety.**

I. Introduction

Detecting the onset of a life-threatening medical emergency in a scuba diver, such as cardiac arrest, is an immense challenge. Underwater, the clear physical symptoms that are obvious on land become obscured by the environment and equipment. While a companion autonomous underwater vehicle (co-AUV) acting as a robotic "dive buddy" is an ideal platform for continuous monitoring [1], [2], [3], [4], communication remains a critical bottleneck. Conventional wireless technologies like WiFi and Bluetooth are highly attenuated in water, making it impossible to stream real-time health data from diver-worn sensors to the AUV. Additionally, lower-frequency acoustic methods simply lack the bandwidth required for immediate inference [5], [6], [7].

This work subverts that fundamental limitation of communicating diver health states. Instead of relying on a data link from the diver, we enable the AUV to use its own vision to detect the most critical sign of severe medical events or disabling injuries (hereafter referred to as *DI*) underwater: *a sudden and complete cessation of movement*. An abrupt transition from swimming to a static, motionless state is a primary indicator that a diver is in distress and requires immediate intervention [8], [9].

To achieve this, we introduce a novel method that transforms the AUV's visual feed into a virtual motion sensor.

Authors [1]Kutzke, [2]Wu, and [4]Sattar are with the Dept. of Computer Science & Engineering and the Minnesota Robotics Institute (MnRI), University of Minnesota–Twin Cities, Minneapolis, MN 55455 USA {[1]kutzk015, [4]junaed}@umn.edu and [2]ethanwu1127@gmail.com

Author [3]Terveen is with the Dept. of Computer Science at Carnegie Mellon University, Pittsburgh, PA 15213 USA [3]eterveen@andrew.cmu.edu

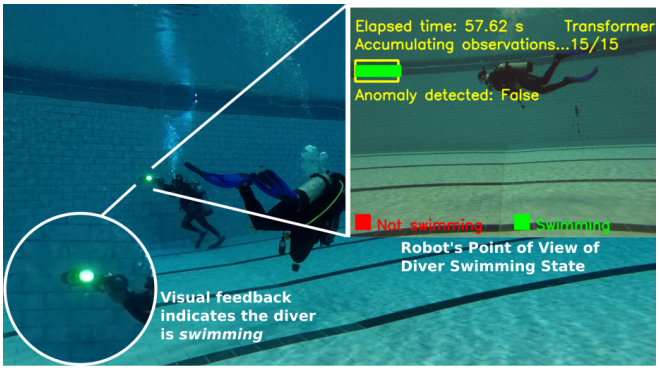


Fig. 1: Temporal classification of scuba diver swimming state conducted during a closed-water evaluation of the diver anomaly classification system. The system is deployed on an AUV. The AUV provides visual feedback of predicted state by illuminating a series of concentric LED lights controlled to reflect a green color which indicates the diver is swimming.

Fig. 1 shows the proposed system during an in-water evaluation of a diver's swimming state. We use three-dimensional human pose estimations from a monocular camera to approximate *translational* acceleration of the diver's joint keypoints. Additionally, we establish a body frame convention to estimate the diver's *rotational* acceleration using the same body keypoints. Together, these constitute a novel approach of using human pose joints for scuba diver swimming state, and eventually health state, classification. By tracking the diver's 3D joint keypoints over time, we generate a stream of translational and rotational acceleration data–effectively creating a "pseudo-inertial measurement unit" (pseudo-IMU) from vision alone. This allows a deep learning model to classify the diver's swimming state in real-time without any instrumentation on the diver.

In this paper, we make two primary contributions: (1) a novel system for classifying a scuba diver's swimming state using only a monocular camera on an AUV, and (2) a unique and diverse dataset of underwater human motion, capturing the transition from swimming to non-swimming states. We demonstrate the effectiveness of our system through in-water experiments, proving it is a viable and powerful new approach for robotic monitoring of diver safety.

II. Literature Review

Our work integrates two primary domains: human pose estimation and time-series classification.

Human Pose Estimation. Human pose estimation is the process of localizing human joints in images, typically using

Deep Neural Networks (DNNs) with convolutional layers to model the topological relationships between limbs [10]. For this work, we require 3D keypoint data to estimate accelerations. 3D pose estimation networks often generate these coordinates by applying triangulation to 2D poses or by training on data with labeled depth information [11], [12], [13]. Performance can be improved by leveraging temporal data from previous frames or through modern attention-based models [14].

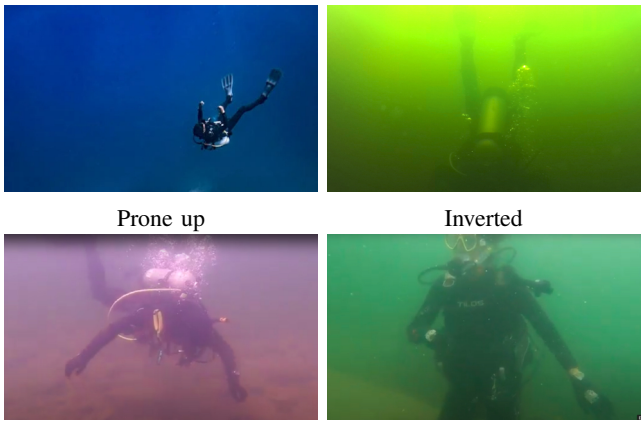


Fig. 2: Typical diver poses during scuba diving operations. Deep neural networks trained on terrestrial data often fail to localize human joints for these poses, given the uniqueness of the joint configurations and visibility.

The underwater environment is a unique challenge, since divers can adopt poses in six degrees of freedom (6DOF) that are highly non-standard in the terrestrial domain (see Fig. 2). Off-the-shelf models trained on terrestrial data often fail to localize joints in these unique configurations. We overcame this by fine-tuning a *YOLOv8* model [15] on a hand-labeled underwater dataset and then using the *VideoPose3D* [16] estimator, trained on AIST++ [17] and H3WB [18], to lift the resulting 2D poses to 3D. While a recent method has emerged for extracting 3D poses underwater using stereo vision without labeled data [19], our approach has the distinct advantage of requiring only a monocular camera.

Time-Series Classification. Time series classification (TSC) [20], [21] assigns a class label to a sequence of data. A related subproblem is time series anomaly detection (TSAD), which identifies anomalous points within a sequence. A collective anomaly [22] refers to the classification of a sequence as anomalous considering the aggregate effect of multiple underlying measurements, even though an individual measurement might not be considered anomalous. Multiple methods exist for detecting collective anomalies, including [23] and [24], both of which utilize discords for anomalous subsequence detection in collective samples, effectively decreasing inference time by ignoring irrelevant dimensions. Anomaly detection is a difficult problem due to the paucity of data available on rare events. This is heightened in settings for which direct measurement of anomalous events is either challenging, impossible, or too rare to capture substantive data. Collecting vital signs of scuba divers during diving operations is such an example, unless conducted under strict experimental settings and with specialized equipment [25].

Though our goal is to detect health anomalies, our method employs TSC rather than TSAD. This allows us to classify an entire sequence as a "swimming state" without needing to precisely engineer the specific features that constitute an anomaly [26]. By collecting data in structured, uniform intervals, we were able to create representative class sequences without manual segmentation, using pseudo-IMU values as swimming state. We followed the method described in [27] to create subsequences of representative data manually, eliminating the need to do feature engineering by hand.

A central challenge in TSC is simultaneously capturing short-term and long-term dependencies. Hybrid models [28], [29] combining Convolutional and Recurrent Neural Networks (CNNs, RNNs) are effective at this, as they merge the strengths of each architecture while mitigating their individual drawbacks (*e.g.*, a CNN's failure to capture long-term patterns or an RNN's computational expense). Attention-based models also excel at capturing long-term dependencies [28], though often at a higher computational cost, especially for embedded systems [30].

III. Data Collection and Feature Extraction

Data collection consisted of two parts: (1) non-standard body pose data collection, which includes collection of images to perform camera calibration to extract camera intrinsic and extrinsic parameters, and (2) diver swimming state transition data from swimming to not swimming. Data was collected in accordance with Institutional Review Board (IRB) regulations and assessed to be not human research.

A Note About the Ethics of In-Water Data Collection. We cannot ask participants to stop moving in open water environments, since swimming, kicking, and paddling are critical for maintaining good buoyancy or depth control in the water column. Without buoyancy control, divers are at risk of significant medical issues such as barotrauma injuries from uncontrolled descents or decompression sickness due to gas embolisms from uncontrolled ascents [31]. To ensure diver safety, we performed data collection and in-water evaluations of our robotic system in a closed-water swimming facility. This allowed us to mitigate risks to the diver as well as decouple the efficacy of the proposed system from environmental conditions. However, in Section VII, we present initial results from ongoing field tests, along with a discussion on planned future improvements, demonstrating the generalization capability of our method to challenging open-water environments.

Non-standard Body Pose Data. Divers can achieve highly atypical body poses underwater compared to the terrestrial domain (see Fig. 2). To ensure that our pose estimation network could accommodate non-standard body poses, we collected and aggregated 3305 stereo pair images of resolution 640×360 pixels at 10 fps; these include divers in four primary poses over different viewpoints: *prone down*, *prone up*, *inverted*, and *upright*. Note that we use a stereo

camera effectively only to double the size of the dataset; our method only uses single images for inference, and camera calibration is used only to remove geometric distortions.

Each diver was asked to rotate 360 degrees about the $\hat{z}$-axis in place while maintaining a given body pose. Divers were asked to do this at distances between 3 and 5 meters from the camera as measured by a trackline with visible markers extending from the camera image plane, along the camera frame's $\hat{x}$-axis. Fig. 3 demonstrates the data collection setup.

Fig. 3: (Best if viewed at 3× zoom level). Experimental setup for in-water data collection of non-standard body pose data and diver state transition data. The trackline delineates distance between the camera and participant. We collected image data at a depth of approximately 3.5 m and between 3 to 5 m from the camera's image plane.

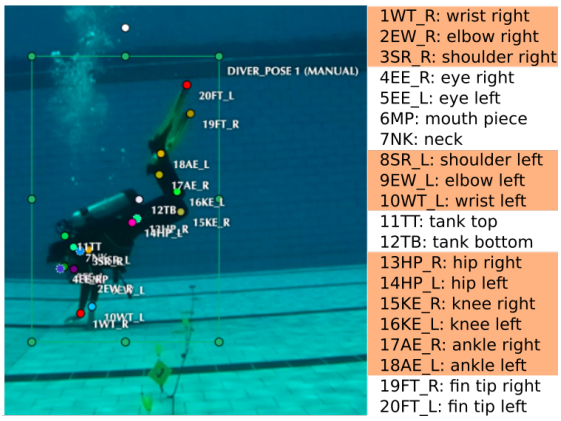
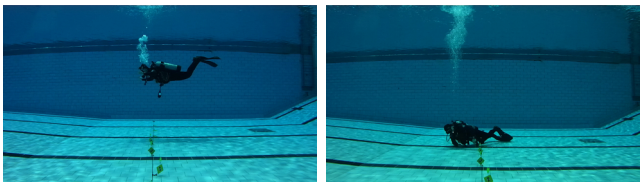


Fig. 4: (Best if viewed at 1.5× zoom level). Pose keypoints labeled using an augmented COCO convention. We utilize the keypoints highlighted in orange in our analysis.

We labeled the images utilizing an augmented COCO convention [32]. We labeled 20 pose keypoints shown in Fig. 4. These keypoints include objects relevant to most recreational divers such as the mouthpiece, top of the tank and bottom, and fin tips. For our analysis, we selected the 12 keypoints corresponding to the diver's major joints (shoulders, elbows, wrists, hips, knees, and ankles), highlighted in Fig. 4 in orange. This set was chosen to capture the dynamics of the limbs and torso, which are the primary drivers of propulsive and stabilizing movements underwater, while excluding less informative points like the eyes or external equipment.

YOLOv8 requires undistorted images for training and inference, and *VideoPose3D* requires a sequence of calibrated two-dimensional poses. To ensure that we obtained accurate pose estimates for our method, we collected three sets of calibration data to compute camera intrinsic and extrinsic matrices. We utilized three different calibration target sizes, including 4×6, 5×8, and 8×10 tags in a grid, using the *Kalibr* target specification [33]. Specific target dimensions are included in the accompanying video.

Diver State Transition Data. Divers were asked to maintain each of the four primary poses (*prone down*, *prone up*, *inverted*, and *upright*) for approximately 10 s by swimming, kicking, or paddling as necessary to maintain that pose. Then they were asked to stop these movements for 5 s. This was done from different viewpoints around the $\hat{z}$-axis as in Fig. 3. By stopping normal swimming activities, this closely mimics the phenomena of sudden cardiac arrest or other DI. Based on the buoyancy adjustment of the participant, sometimes the participant would sink, rise to the surface, or remain in place. Fig. 5 shows one case in which the diver sank to the bottom during the no movement phase of data collection.



Swimming Not swimming

Fig. 5: Example moving and not moving images. In this case, the diver is hovering *prone down*, and when they stop moving they sink.

Feature Extraction. The cornerstone of our method is the conversion of raw 3D keypoint coordinates into a meaningful, motion-based feature vector. This process involves two key steps: first, estimating up-to-scale 3D keypoints from divers using off-the-shelf estimators, and second, establishing a stable body-centric reference frame to compute the diver's overall translational and rotational acceleration.

Let the input to the pose estimation network be an undistorted image of width W, height H, and resolution $W \times H$. Let $\mathbf{K} = \{\mathbf{k}_1, \ldots, \mathbf{k}_M\}$ be the set of predicted human joint keypoints in image coordinates, *i.e.*, $\mathbf{k} = (u, v) \in [0, W-1] \times [0, H-1]$. Note that we utilize $M = 12$ different keypoints defined in Fig. 4. The 2D-to-3D lifting network then transforms these predictions into 3D space with arbitrary scale. Now let $\mathbf{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_M\}$ be the set of corresponding human joint keypoints in three-dimensional coordinates with respect to the camera frame, *i.e.*, $\mathbf{r} = (x, y, z) \in \mathbb{R}^3$. One can show that using a Taylor series expansion, the *translational acceleration* in the x-direction $\ddot{x}$ is approximately

$$\ddot{x} \approx \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2}, \tag{1}$$

where Δt is the temporal separation of image frames, which is effectively the inverse of the frame rate of the camera. x_i is the measurement of the x position at time step i. Similar formulas exist for the $\hat{y}$- and $\hat{z}$-directions.

To accurately measure the diver's self-initiated rotation, we must first decouple their movements from the motion of the AUV's camera. We achieve this by defining a dynamic body-centric reference frame affixed to the diver's torso (see

Fig. 6). This ensures our rotational acceleration features are invariant to the robot's position and orientation. The frame convention relies on the torso pose keypoints of the diver $\mathbf{r}_{\text{torso}} \in \mathbf{R}$, creating a frame that is located at the approximate center of the human's chest. Let $\mathbf{r}_{\text{torso}} = \{\mathbf{r}_{\text{left hip(lh)}}, \mathbf{r}_{\text{right hip(rh)}}, \mathbf{r}_{\text{left shoulder(ls)}}, \mathbf{r}_{\text{right shoulder(rs)}}\}$. Then,

1) We compute the center of the predicted keypoints as $\mathbf{r}_{\text{o}} = \langle \mathbf{r}_{\text{torso}} \rangle$, where $\langle \cdot \rangle$ defines the vector average computation. The resultant vector $\mathbf{r}_{\text{o}}$ is located approximately at the center of the diver's torso.
2) We define several difference vector quantities that exist on the torso plane as

$$\mathbf{r}_{\text{rsrh}} = \mathbf{r}_{\text{rs}} - \mathbf{r}_{\text{rh}} \qquad \mathbf{r}_{\text{lsrh}} = \mathbf{r}_{\text{ls}} - \mathbf{r}_{\text{rh}} \tag{2}$$

$$\mathbf{r}_{\text{lslh}} = \mathbf{r}_{\text{ls}} - \mathbf{r}_{\text{lh}} \qquad \mathbf{r}_{\text{rslh}} = \mathbf{r}_{\text{rs}} - \mathbf{r}_{\text{lh}}. \tag{3}$$

   These quantities are needed to establish the relationships between joint locations, effectively defining the torso plane and conditioning the proceeding analysis with respect to the torso plane.
3) We compute the diver's facing direction by taking the average direction of the cross product between the difference vectors of the torso joints. This defines a direction perpendicular to the torso plane

$$\mathbf{r}_{\text{r}_\times} = \mathbf{r}_{\text{lsrh}} \times \mathbf{r}_{\text{rsrh}} \tag{4}$$

$$\mathbf{r}_{\text{l}_\times} = \mathbf{r}_{\text{lslh}} \times \mathbf{r}_{\text{rslh}}. \tag{5}$$

   To compute the average direction and define a unit vector, we first take the average, and then we divide by the vector $L2$-norm

$${}^{c}\hat{z}_B \equiv \frac{\langle \mathbf{r}_{\text{r}_\times}, \mathbf{r}_{\text{l}_\times} \rangle}{\| \langle \mathbf{r}_{\text{r}_\times}, \mathbf{r}_{\text{l}_\times} \rangle \|_2}. \tag{6}$$

   The alignment vector given by (6) points in a direction perpendicular to the plane defined by the torso keypoints. We now affix a right-handed coordinate system to $\mathbf{r}_{\text{o}}$, with ${}^{c}\hat{z}_B$ aligned along the direction given in (6). We choose ${}^{c}\hat{y}_B$ to be the vector that points along the direction of the midpoint between hip joints. This is given by computing the midpoint of the line segment connecting the hip joints

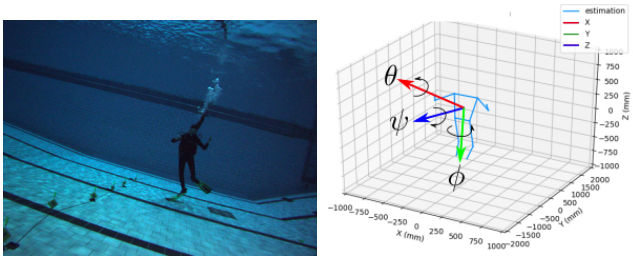
$$\mathbf{r}_{\text{midpt}} = \langle \mathbf{r}_{\text{lh}}, \mathbf{r}_{\text{rh}} \rangle. \tag{7}$$

4) From this we compute the unit vector that points from the center of mass vector $\mathbf{r}_o$ to $\mathbf{r}_{\text{midpt}}$. This unit vector is defined to be ${}^{c}\hat{y}_B$

$${}^{c}\hat{y}_B = \frac{\mathbf{r}_{\text{midpt}} - \mathbf{r}_o}{\|\mathbf{r}_{\text{midpt}} - \mathbf{r}_o\|_2}. \tag{8}$$

5) Finally, the ${}^{c}\hat{x}_B$ is computed through a cross product ${}^{c}\hat{x}_B = {}^{c}\hat{y}_B \times {}^{c}\hat{z}_B$. Together these constitute the body frame ${}^{c}\mathcal{F}_B = [{}^{c}\hat{x}_B, {}^{c}\hat{y}_B, {}^{c}\hat{z}_B, \mathbf{r}_{\text{o}}]$ of the human diver, affixed to the midpoint of the extracted pose keypoints, with the ${}^{c}\hat{z}_B$ aligned in the direction perpendicular to the plane defined by the torso keypoints.

Rotational acceleration of the body frame is an approximation to the second derivative for each of the rotational angles



Raw image (brightened for visibility). Body frame showing the rotation angles.

Fig. 6: (Best if viewed at 2× zoom level). Raw input image and the body frame with rotation angles after three-dimensional pose estimation.

about the unit vectors of ${}^{c}\mathcal{F}_B$. One can consult [34] for a detailed derivation of the second derivative of a rotation matrix. For brevity, we utilize the notation θ to indicate the rotation angle about the $\hat{x}$-axis, ϕ to indicate the rotation angle about the $\hat{y}$-axis, and ψ to indicate the rotation angle about the $\hat{z}$-axis. Fig. 6 shows this convention.

Now let $\mathbf{X} = \{\mathbf{x}_i\}$, where $i = \{1, \ldots, N\}$ and N is the total number of time steps, define the feature vector. We then define a feature vector for a single time step as

$$\mathbf{x}_i = \{\ddot{x}_{i,1}, \ddot{y}_{i,2}, \ddot{z}_{i,3}, \ldots, \ddot{x}_{i,M-1}, \ddot{y}_{i,M-1}, \ddot{z}_{i,M-1}, \ddot{\theta}_i, \ddot{\phi}_i, \ddot{\psi}_i\}, \tag{9}$$

where the first $1, \ldots, M-1$ terms represent approximations to the second derivatives with respect to three-dimensional positions for the $M = 12$ keypoints, and the last three terms are the second derivative with respect to the rotation frame of the human body. To remove the influence of the camera movement, we subtract the left hip keypoint location ($\mathbf{r}_{\text{lh}}$) from all keypoints. Therefore every keypoint is located with respect to the left hip. We then remove the left hip keypoint acceleration from the feature vector, effectively reducing the dimensionality to $N \times (3(M-1)+3)$ for both translational and rotational feature vectors. Along with horizontal image flipping and random rotation on 3D diver pose sequence estimates, this feature extraction pipeline is used to create a train-test dataset.

IV. Methodology

The diver anomaly classification system described in this paper relies on two subsystems: a feature extractor (described in Sec. III), which extracts pseudo-IMU values, and a classification system to perform inference based on the temporal observations of the diver's state over time. Fig. 7 shows a summary of the methodology, where the top left block demonstrates the input image sequence, from which *YOLOv8* produces two-dimensional pose estimates. *VideoPose3D* requires 27 two-dimensional pose estimates to produce a single three-dimensional estimate. After $N+2$ three-dimensional poses, which are required to estimate the acceleration quantities of the keypoints and body frame using *central difference approximations*, we construct the feature vector. The classifier (shown in yellow in Fig. 7) uses a feature vector of size $N \times (3(M-1)+3)$, $N \times 3$, or $N \times$

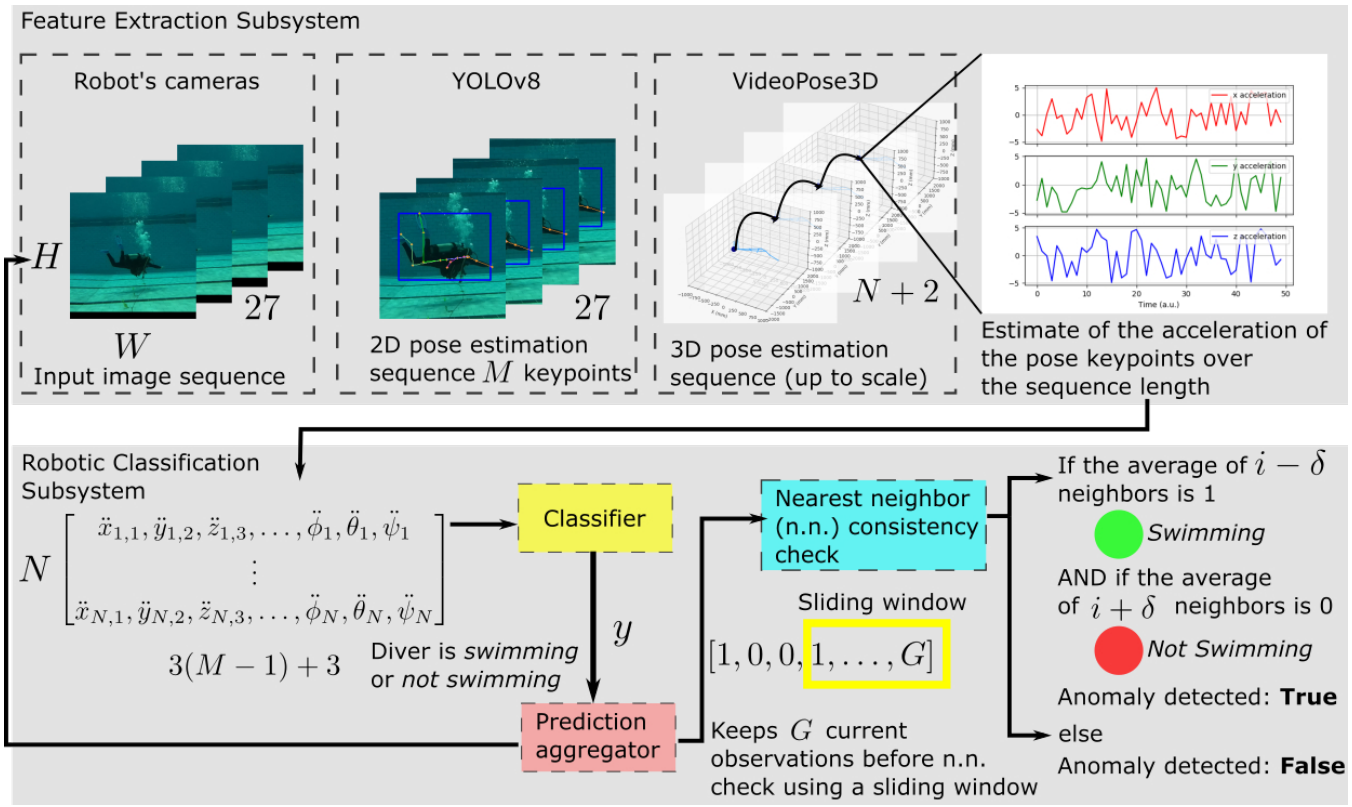


Fig. 7: (Best if viewed at 2.5× zoom level). Overview of the diver anomaly classification system. The system comprises two functional subsystems: a feature extractor, which creates the pseudo-IMU vector, and the robotic classifier that utilizes an arbitrary classifier for determining if the diver is *swimming* or *not swimming*.

$(3(M-1))$, for combined rotation and acceleration, rotation only, or translation only acceleration features, respectively, to classify the sequence as either *swimming* or *not swimming*.

The prediction aggregator (shown in salmon color in Fig. 7) aggregates these observations. To mitigate erroneous observations, we employ a nearest-neighbor consistency check (shown in blue in Fig. 7), which utilizes a sliding window over the latest G observations. If there exists some prediction y_j, $j \in \{1, \dots G\}$ for which the average of the past δ neighbors is 1 and the average of the future δ neighbors is 0, then the diver has transitioned from *swimming* to *not swimming*. We empirically determined that $\delta = 7$, and $G = 15$ produce higher state transition accuracy.

The primary contribution of this work is an approach to swimming state transition classification that leverages visual changes in body keypoints as a proxy for IMUs to assess diver pose. To thoroughly evaluate the effectiveness of our pseudo-IMU features, we benchmarked them using a diverse suite of six time-series classification (TSC) models. This suite was chosen to represent a range of approaches, from established statistical methods to state-of-the-art deep learning architectures. We utilize six different methods for TSC: time series forest based on [35], CNN and CNN Channel Wise based on [36], CNN LSTM Dual Network from [37], CNN LSTM Layer Wise from [38], and Transformer based on [39]. We describe each method briefly below but refer the reader to the relevant literature source for more detailed explanations and network architectures.

The **Time Series Forest** serves as a non-neural baseline to evaluate the inherent discriminative power of our features without complex deep learning. It utilizes $N_{\text{trees}} = 500$ trees for splitting the pseudo-IMU feature dataset nodes into the trees. A splitting criterion based on entropy gain is used to decide node splits within the tree. Additionally, [35] introduces the concept of the margin split, which helps reduce the number of candidate subsequence splits that produce the same entropy gain.

CNN and CNN Channel Wise [36] introduces both a CNN-based classification architecture, which utilizes a series of stages consisting of layers of convolution, activation, and pooling layers, and a CNN Channel Wise (CNN CW) method that splits the time series into individual univariate subsequences, performs feature extraction on each subsequence, concatenates the output, and then follows with multilayer perceptron classification. Both of the above CNN-based models were chosen to test the hypothesis that local patterns of acceleration (*e.g.*, the kick cycle) are the most important features for classification.

The **CNN LSTM Dual Network** [37] (CNN LSTM DN) uses a CNN-based architecture for feature extraction, and then it applies an LSTM in parallel. The resulting output from both the CNN and the LSTM are concatenated and flattened for classification. We employ a notable modification from the original network architecture: unlike [37] who use

three convolutional blocks, we achieved sufficient results with one convolutional block on both open-source datasets and underwater-specific data.

CNN LSTM Layer Wise [38] (CNN LSTM LW) architecture, in contrast to the (CNN LSTM DN) first uses convolutional layers for feature extraction and then applies an LSTM on the extracted features, rather than the raw input sequence itself. This model was included to investigate whether combining local feature extraction (CNN) with longer-term temporal dependency modeling (LSTM) would yield superior performance.

The **Time-Series Transformer** introduced in [39], was included to determine if the global self-attention mechanism, which can relate data points across the entire 50-step sequence, could capture complex, long-range motion dynamics that other models might miss. Although the original paper employs *fixed* positional encodings, subsequent studies have demonstrated that *learnable* positional encodings yield better results [40]. As such, we project the time series onto an embedding space using a matrix with learnable weights before feeding it to a self-attention block. Additionally, while layer normalization is optimal for natural language processing, batch normalization has been shown to yield better performance when applied to numerical input [41]. Therefore, within the self-attention block, we substitute batch normalization for the layer normalization layers employed by [42]. Following [39], we begin the model training process with an unsupervised pre-training stage, where the input series is randomly masked and the model attempts to predict masked target values. This approach helps the network learn the shape of the data, facilitating speed-ups in the supervised learning phase. Pre-training is particularly useful given our constraints – transformers demand large quantities of training data to learn representations of input information. Visual information is significantly more costly to collect and label than hardware-based IMU data, a bottleneck intensified by the unusually high requirements of underwater equipment. This limits the quantity of visual training data available, which we significantly mitigate through pre-training.

V. Classification Method Evaluation

We evaluated each classification method offline using 6114 train, 1528 validation, and 1713 test samples, with $N = 50$ time steps, on an NVIDIA GeForce RTX 2080 Ti GPU with PyTorch 2.0.1, CUDA 11.7, and cuDNN 8. Table I shows the classification accuracies comparing different features. We evaluated *translational* only, shown in the first column of the table; *rotational* only, shown in the second column; and combined *rotational* and *translational* features, shown in the last column of the table. Results revealed that *translational* features produced the highest accuracies across all methods. The CNN channel-wise method also performed the highest for individual features. However, the time series forest achieved 90.83 on the combined features, exceeding the CNN channel wise method. Translational features provide superior accuracy because they are inherently more robust to the "jitter" noise present in the pose estimation

TABLE I: Comparison of classification accuracies utilizing translational, rotational, and combined acceleration features. We utilized 6114 train, 1528 validation, and 1713 test samples for our model training and evaluation pipeline, with $N = 50$ three-dimensional pose estimates used for computing the acceleration data in the feature vector.

Bench test classification accuracy statistics (% correct)[a]			
Method	**Trans. features** (50×33)	**Rot. features** (50×3)	**Rot.+Trans. features** (50×36)
Time Series Forest [35]	90.13	71.63	**90.83**
CNN [36]	89.32	76.71	89.84
CNN Channel Wise [36]	**91.30**	**85.99**	90.37
CNN LSTM Dual Network [37]	82.60	52.95	82.95
CNN LSTM Layer Wise [38]	89.20	79.92	85.93
Transformer[b] [39]	90.83	65.38	84.18

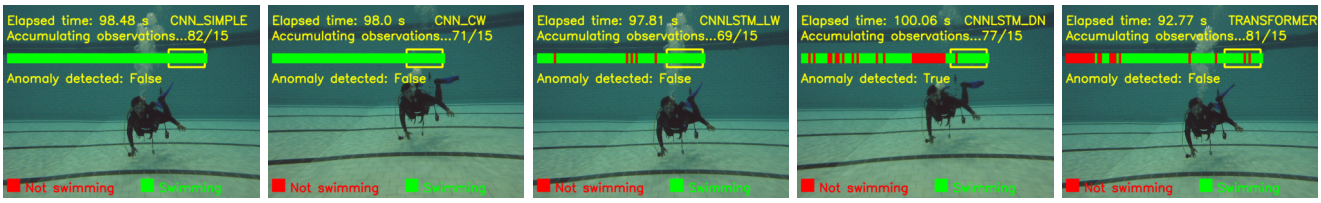
[a] All methods were trained using 50 epochs.
[b] The transformer is pre-trained for 300 epochs using the method described in [39] and fine-tuned for 50 epochs on our data.

data, whereas rotational accelerations, in contrast, are highly sensitive to this noise. Their calculation depends on a rotation axis derived from just four torso keypoints $\mathbf{r}_{\text{torso}}$; if these points are noisy, the axis becomes unstable and corrupts the final feature. Conversely, our translational features are derived from the mean motion of all 12 body keypoints. This averaging provides a powerful smoothing effect that filters out the noise from individual joints, resulting in a stable and reliable measure of the diver's aggregate movement. Consequently, when a diver is motionless, the translational acceleration correctly approaches zero, providing a much cleaner signal for classification. Additionally in Sec. VII, experimental results demonstrate how the proposed method can generalize across multiple underwater domains with the aid of a robust pose estimator (*e.g.*, [19]).

VI. Closed-water Evaluation

We performed in-water evaluations of the diver swimming state transition system by deploying the system onboard an AUV that uses an Nvidia Jetson TX2 embedded computing GPU. We built our codebase in a docker image [43], with Python3 and ROS Noetic.

We evaluated five classification methods with an experimental setup similar to the non-standard body pose data collection event. The diver swam in front of the robot's camera in a prone down position (the most common form of pose employed by divers, since it is an efficient swimming posture). The diver swam until signaled to stop. This ensured that the feature extractor subsystem acquired sufficient observations to produce the first feature vector required for classification. The diver then stopped all movement for 50s. We evaluated each classifier based on its ability to classify the diver as swimming and not swimming. We utilized a nearest-neighbor distance $\delta = 7$ and required $G = 15$ classifications before inferring if the diver experienced a transition from swimming to not swimming. Fig. 8 demonstrates the results taken from five of the six classification methods. Due to a software integration conflict, we were unable to deploy our time series forest classification method on the



CNN CNN CW CNN LSTM LW CNN LSTM DN Transformer

Fig. 8: (Best if viewed at $4\times$ zoom). Closed-water evaluation of the swimming state classification methods.

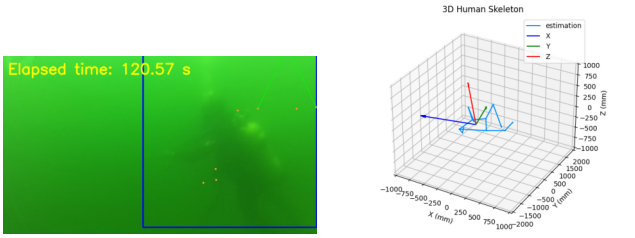
robot's TX2. Figures display the time elapsed from the start of the classification ROS node to the publication time of the feature vector topic. Differences in elapsed time are a product of inference time for each method. The display shows $count/G$, meaning that the first G classifications are required before performing a nearest-neighbor check using a sliding window. The sliding window has a width of δ and helps us visualize the nearest-neighbor check process. Fig. 8 shows the sliding window as a yellow bordered rectangle, which is co-linear with the observation state diagram, shown as a series of rectangles, either green or red, for *swimming* or *not swimming* classification, respectively. The accompanying video submission shows the sliding window moving as additional observations accumulate.

We utilized translational features, since bench experiments revealed translational features offer the highest testing accuracy across all methods. Although the CNN CW performed best during bench testing, with an accuracy of 91.30% (shown in Table I), the CNN LSTM DN performed the best during in-water evaluation, accurately determining when the diver transitioned from *swimming* to *not swimming*, approximately 45 s into the evaluation experiment. Additionally, the AUV is equipped with a series of LED ring lights that communicate visual feedback of onboard processes. The AUV illuminated the lights in a green color when the diver was *swimming*. Fig. 1 shows these lights illuminated.

VII. Open-Water Field Evaluation

We evaluated our pipeline on previously unseen data collected from a freshwater environment, in which two divers operated at a depth of five meters in approximately two meters of visibility. The water was turbid and green in color, which resulted in one of the worst visibility conditions. The camera subject was asked to perform transitions from *swimming* to *not swimming*. The *swimming* portion of the data collection was 2 minutes, and the *not swimming* was 30 s. The results for the five primary networks are shown in Fig. 10. Notice that in all cases, the classification networks observed *not swimming* states during the time period that the diver maintained a prone facing down position. This swimming behavior resulted in very little arm and leg movement, which could indicate why the classification methods failed to classify this as a *swimming* state.

These preliminary results also prove our hypothesis that the accuracy of the pose estimation network has downstream effects on our method for detecting anomalous swimming behavior. Fig. 9 shows that the two-dimensional estimation



2D pose estimation 3D reconstructed human pose

Fig. 9: Pose estimation figures from the 2D pose estimation and resulting 3D reconstructed human pose keypoints.

both fails to localize the human in the image frame and to locate the human pose keypoints on the diver's torso. These results show that the pose estimation network does not transfer between environmental conditions, since the pose network was trained on closed-water swimming pool data.

VIII. Conclusion

We introduce a novel system for robotic classification of a diver's swimming state, which is able to detect when a diver transitions from normal movement, such as kicking or paddling, to no movement. This mimics the state of a diver during a DI, which could lead to adverse health effects or a loss of life. While arrested motion is not the exclusive indicator of scuba diver distress, this work makes it possible to assess diver motion characteristics without requiring on-body sensors, while avoiding underwater data transmission challenges. Our ongoing work is investigating a multimodal assessment approach of diver distress using additional DI indicators such as respiration rate, along with field trials in diverse regions to assess its efficacy.

References

[1] M. J. Islam, M. Ho, and J. Sattar, "Understanding human motion and gestures for underwater human-robot collaboration," *Journal of Field Robotics*, vol. 36, no. 5, pp. 851–873, 2019.

[2] R. Codd-Downey and M. Jenkin, "Recognizing diver hand gestures for human to robot communication underwater," in *Proceedings of the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2023, pp. 92–98.

[3] A. Birk, "A survey of underwater human-robot interaction (U-HRI)," *Current Robotics Reports*, vol. 3, pp. 199—211, 2022.

[4] K. J. DeMarco, M. E. West, and A. M. Howard, "Underwater human-robot communication: A case study with human divers," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 3738–3743.

[5] R. V. Lundell, A. K. Räisänen-Sokolowski, T. K. Wuorimaa, T. Ojanen, and K. I. Parkkola, "Diving in the arctic: Cold water immersion's effects on heart rate variability in Navy divers," *Frontiers in Physiology*, vol. 10, p. 1600, 2020.

CNN | CNN CW | CNN LSTM LW | CNN LSTM DN | Transformer

Fig. 10: (Best if viewed at $4\times$ zoom). Open-water evaluation of the swimming state classification methods.

[6] G. Bosco, A. Rizzato, R. E. Moon, and E. M. Camporesi, "Environmental physiology and diving medicine," *Frontiers in Psychology*, vol. 9, p. 72, 2018.

[7] K. Dimmock and E. Wilson, "Risking comfort? The impact of in-water constraints on recreational scuba diving," *Annals of Leisure Research*, vol. 12, no. 2, pp. 173–194, 2009.

[8] P. Wilmshurst and C. M., "Impaired consciousness when scuba diving associated with vasovagal syncope," *Diving & Hyperbaric Medicine*, vol. 4, no. 50, pp. 421–423, December 2020.

[9] M. Di Paolo, E. Mezzetti, M. Leoni, A. Scatena, and C. Passino, "Sudden cardiac death during scuba diving: a case report of a patient with unknown hypertrophic cardiomyopathy," *European Heart Journal–Case Reports*, vol. 8, no. 5, May 2024.

[10] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–37, 2023.

[11] K. Lee, I. Lee, and S. Lee, "Propagating LSTM: 3D pose estimation based on joint interdependency," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 119–135.

[12] J. C. Núñez, R. Cabido, J. F. Vélez, A. S. Montemayor, and J. J. Pantrigo, "Multiview 3D human pose estimation using improved least-squares and LSTM networks," *Neurocomputing*, vol. 323, pp. 335–343, 2019.

[13] J. Wang, S. Tan, X. Zhen, S. Xu, F. Zheng, Z. He, and L. Shao, "Deep 3D human pose estimation: A review," *Computer Vision and Image Understanding*, vol. 210, p. 103225, 2021.

[14] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3D human pose estimation with spatial and temporal transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 11 656–11 665.

[15] Ultralytics, "Yolov8," Ultralytics, Tech. Rep., 2023.

[16] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7753–7762.

[17] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, "AI choreographer: Music conditioned 3D dance generation with AIST++," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 401–13 412.

[18] Y. Zhu, N. Samet, and D. Picard, "H3WB: Human3.6M 3D whole body dataset and benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 20 166–20 177.

[19] Y.-K. Wu and J. Sattar, "Stereo-based 3D human pose estimation for underwater robots without 3D supervision," *IEEE Robotics and Automation Letters*, 2025.

[20] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proceedings of the VLDB Endowment*, vol. 15, no. 9, pp. 1779—-1797, may 2022.

[21] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[22] L. Bontemps, V. L. Cao, J. McDermott, and N.-A. Le-Khac, "Collective anomaly detection based on long short-term memory recurrent neural networks," in *Proceedings of the International Conference on Future Data and Security Engineering (FDSE)*. Springer, November 2016, pp. 141–152.

[23] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently finding the most unusual time series subsequence," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2005, p. 8.

[24] C.-C. M. Yeh, Y. Zheng, M. Pan, H. Chen, Z. Zhuang, J. Wang, L. Wang, W. Zhang, J. M. Phillips, and E. Keogh, "Sketching multidimensional time series for fast discord mining," in *Proceedings of the IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 443–452.

[25] B. M. Keuski, "Updates in diving medicine: Evidence published in 2017–2018," *Undersea & Hyperbaric Medicine*, vol. 45, no. 5, 2018.

[26] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C.-S. Foo, "An evaluation of anomaly detection and diagnosis in multivariate time series," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2508–2517, 2021.

[27] B. Hu, Y. Chen, and E. Keogh, "Classification of streaming time series under more realistic assumptions," *Data Mining and Knowledge Discovery*, vol. 30, pp. 403—-437, 2016.

[28] N. Mohammadi Foumani, L. Miller, C. W. Tan, G. I. Webb, G. Forestier, and M. Salehi, "Deep learning for time series classification and extrinsic regression: A current survey," *ACM Computing Surveys*, vol. 56, no. 9, April 2024.

[29] H. Ismail Fawaz, G. Forestier, J. Weber, I. Lhassane, and M. Pierre-Alain, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, pp. 917–963, 2019.

[30] S. Mukhopadhyay, S. Dey, A. Mukherjee, A. Pal, and S. Ashwin, "Time series classification on edge with lightweight attention networks," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2024, pp. 487–492.

[31] C. Carlston, R. Mathias, and C. Shilling, *The physician's guide to diving medicine*. Springer Science & Business Media, 2012.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.

[33] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1360–1367.

[34] H. Goldstein, C. Poole, and J. Safko, *Classical mechanics*, 3rd ed. Addison Wesley, 2002.

[35] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A time series forest for classification and feature extraction," *Information Sciences*, vol. 239, pp. 142–153, 2013.

[36] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channel deep convolutional neural networks," in *Proceedings of the International Conference on Web-Age Information Management*. Springer, 2014, pp. 298–310.

[37] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[38] R. Mutegeki and D. S. Han, "A CNN-LSTM approach to human activity recognition," in *Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 362–366.

[39] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2114–2124.

[40] A. Haviv, O. Ram, O. Press, P. Izsak, and O. Levy, "Transformer language models without positional encodings still learn positional information," *arXiv preprint arXiv:2203.16634*, 2022.

[41] Z. Yao, Y. Cao, Y. Lin, Z. Liu, Z. Zhang, and H. Hu, "Leveraging batch normalization for vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 413–422.

[42] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[43] D. Merkel, "Docker: Lightweight Linux containers for consistent development and deployment," *Linux Journal*, vol. 239, no. 2, p. 2, 2014.